**INT232 PROJECT REPORT**

(Project 4th Semester January-May 2024)

**DASHBOARD ON SALES OF COFFEE STORE CHAIN**

Submitted by

Sai Sasank Lakimsetti

Registration No. – 12209302

Program – Computer Science and Engineering

Section – K22CA

Course Code – INT232

Under the guidance of

**Zeenat Zahra: 30447**

**Discipline of CSE/IT**

**Lovely School of Computer Science and Engineering**

**Lovely Professional University, Phagwara**

**CERTIFICATE**

This is to certify that **Sai Sasank Lakimsetti** bearing Registration no. 12209302 has completed the INT232 project titled, “Dashboard on Sales on Coffee Store Chain” under my guidance and supervision. To the best of my knowledge, the present work is the result of his/her original development, effort, and study.

**Signature and Name of the Supervisor**

**Designation of the Supervisor**

**School of Computer Science and Engineering**

Lovely Professional University

Phagwara, Punjab

Date: 19th April 2024

**DECLARATION**

I, Sai Sasank Lakimsetti, student of Bachelor of Technology under CSE/IT discipline at, Lovely Professional University, Punjab, hereby declare that all the information furnished in this project report is based on my intensive work and is genuine.

Date: 19th April 2024 Signature

Registration No. – 12209302 Sai Sasank Lakimsetti

**Acknowledgment**

I would like to thank Ms. **Zeenat Zahra** for her vital cooperation and help in ensuring the successful completion of my project. Her valuable guidance has been the one that helped me patch this project. Her valuable guidance has been the one that helped me patch this project and make it a full proof success.

Secondly, I’d like to thank Lovely Professional University for giving me this golden opportunity to learn many things, to learn another aspect of life.

Finally, I want to convey my sincere thanks to my family and friends for their constant support and reassurance.

Sai Sasank Lakimsetti

12209302

**Table of Contents**

|  |  |
| --- | --- |
| **Title** | **Page** |
| Cover Page | 1 |
| Certificate | 2 |
| Declaration | 3 |
| Acknowledgment | 4 |
| Table of Contents | 5 |
| Introduction | 6 |
| Objectives | 7 |
| Existing System | 8 |
| Source of Dataset | 9 |
| Packages or Libraries used | 10-11 |
| Analysis on Dataset | 12-20 |
| Final Dashboard | 21-25 |
| Future Scope and Summary | 26 |
| References | 27 |

**INTRODUCTION**

This report dives into the sales performance of a Coffee Store Chain for the first half of 2023, leveraging the power of R Markdown and R programming for data analysis and visualization.

The interactive dashboard presented here offers a comprehensive view of key sales metrics, including quantity of products sold, profit on products sold, and sales trends over time.

While this report utilizes R Markdown and R programming for data exploration, alternative approaches for creating dashboards include Business Intelligence (BI) Tools, Tableau, Programming languages like Python, etc. These tools give users the ability to collect, examine, and present data from many sources in an interactive dashboard that may be accessed online or on a mobile device.

This report empowers stakeholders with valuable insights into a Coffee Store Chain’s sales performance during the first half of 2023. By leveraging the data presented, we can make informed decisions to enhance customer satisfaction, optimize inventory management, and ultimately drive business growth.

**OBJECTIVES**

1. Store Sales Analysis: This objective focuses on analyzing sales performance across different stores based on their locations. By comparing sales figures, you can identify which stores are performing well and potentially uncover factors contributing to their success.

2. Product Category Sales Analysis: This objective involves analyzing sales data based on product categories. By examining the sales figures for each category, you can identify top-performing categories and potentially adjust inventory or marketing strategies accordingly.

3. Product Type Sales Analysis: Similar to objective 2, this objective involves analyzing sales data based on product types. By examining sales figures for each product type, you can gain insights into customer preferences and determine which types of products are driving sales.

4. Quantity Analysis: This objective focuses on analyzing sales data based on the quantity of products sold. By examining sales figures in terms of quantity, you can identify trends in product demand and potentially adjust pricing or promotions to drive sales.

5. Temporal Analysis: This objective involves analyzing sales data based on purchase date and time. By examining sales trends over time, you can identify peak sales periods, seasonal variations, and other temporal patterns that may influence purchasing behavior.

**Drawbacks or limitations of the existing system**

R is a powerful tool for data analysis and visualization, but it does have some limitations to consider:

* Steep Learning Curve: R’s syntax and package ecosystem can be challenging for beginners, potentially extending the time needed to create the dashboard.
* Performance: R’s interpreted nature can lead to slower processes compared to compiled languages like Python, particularly when dealing with large datasets. This might impact the interactivity of your dashboard.
* Memory Management: R’s memory management can be less efficient than other languages. When working with extensive data, we might encounter memory limitations that could have affected the size and complexity of the visualizations.
* Data Security Concerns: As an open-source language, R might raise security concerns when handling sensitive sales data, especially in commercial environments. For situations involving sensitive information, languages with stricter security protocols or additional security measures might be necessary.
* Lack of Commercial Support: Since R is open-source, there’s no single entity providing official commercial support. While there’s a large and active R community, for troubleshooting complex issues, commercially supported languages might offer more reliable assistance.
* Limited 3D Graphics Capabilities: R’s heritage from the S language can limit its ability to produce advanced 3D visualizations. While it offers 3D plotting functionalities, they might not be as sophisticated or user-friendly as those found in other languages like Python’s libraries. This could restrict your ability to create highly immersive or complex 3D visualizations in your dashboard, even if the data itself might benefit from such a representation.
* Scattered Functionality across packages: R’s extensive package ecosystem offers a vast array of functionalities for data analysis and visualization. However, this can also lead to fragmentation, where desired functionalities might be spread across multiple packages. This can make the process of finding and integrating the right tools for your specific data analysis and visualization tasks more time-consuming compared to languages with centralized or streamlined functionality.
* Steep learning curve for visualization: R’s base graphics capabilities can be considered less intuitive compared to some other data visualization libraries. While powerful, creating polished and customized visualizations in R often requires learning additional packages like ggplot2, which adds another layer to the learning curve for those new to data visualization.

These limitations can be mitigated by:

* Data preprocessing and optimization: By efficiently structuring and cleaning data before analysis, you can minimize the performance impact of large datasets in R.
* Leveraging advanced visualization libraries: Exploring powerful libraries like ggplot2 can unlock a wider range of customization options and create more visually appealing dashboards within R.
* By utilizing third-party packages or by integrating R Studio with other software tools. Furthermore, for many users, the advantages of R Studio—such as its adaptability and capacity for handling sophisticated statistical analyses—often exceed these drawbacks.

**Source of dataset**

The dataset used is taken from Kaggle. Kaggle is a platform that hosts a variety of datasets from different domains such as healthcare, finance, sports, and more. The datasets on Kaggle are contributed by users and organizations from all over the world.

To access datasets on Kaggle, you first need to create an account on the platform. Once you have an account, you can search for datasets using the search bar on the Kaggle homepage or browse through the datasets by category.

Some of the popular datasets on Kaggle include the Titanic dataset, the IMDB movie review dataset, the New York City Airbnb dataset, and Lionel Messi vs Cristiano Ronaldo | Club Goals datasets.

The dataset used has daily-level information on the sales of products across stores in the first half of the year 2023.

Details of the dataset chosen:

* Name: Coffee Shop Sales Analysis
* Link: <https://www.kaggle.com/datasets/divu2001/coffee-shop-sales-analysis>
* Format: CSV
* No. of datasets: 1
* Number of rows: 149117
* Number of columns: 18
* Size: 17.54 MB
* Data Fields:

1. Transaction id
2. Transaction date
3. Transaction time
4. Store id
5. Store location
6. Production id
7. Transaction Quantity
8. Unit price
9. Total bill
10. Product category
11. Product type
12. Product Detail
13. Size
14. Month name
15. Day name
16. Hour
17. Month
18. Day of the week

Note: Column names might be different in the actual CSV file. Names have been changed above for better understanding.

**Packages or Libraries used**

1. **Flexdashboard**

Flexdashboard is a powerful R package built on top of R Markdown that simplifies the creation of interactive dashboards. Here’s a quick rundown of its key features:

* + Easy layout creation: Flexdashboard allows you to define layouts using a grid system, making it straightforward to arrange various visualizations and text elements on your dashboard. This ensures a visually organized and user-friendly presentation of your sales data.
  + Supports diverse content: Flexdashboard integrates seamlessly with various data visualization tools in R. You can incorporate charts created with base R graphics, lattice, ggplot2, or even htmlwidgets for interactive elements. This flexibility allows you to represent your data using the most appropriate and informative visualizations.
  + Mobile-responsive design: Flexdashboard dashboards are designed to adapt automatically to different screen sizes, ensuring optimal viewing experience on desktops, tablets, and mobile devices. This is crucial for ensuring your insights are accessible to users regardless of their device.
  + Integration with Shiny: For more complex interactivity, Flexdashboard integrates with the Shiny package. This allows you to incorporate user input elements like sliders or dropdown menus, enabling viewers to explore the data and visualizations dynamically within your dashboard. This can be particularly useful for allowing stakeholders to filter the data by specific criteria.
  + Markdown for Rich Text: Flexdashboard leverages R Markdown, allowing you to include rich text descriptions and explanations alongside your visualizations. This empowers you to provide context and insights into the data presented on the dashboard.

Overall, Flexdashboard offers a user-friendly and versatile framework for creating interactive dashboards in R. It streamlines the process of presenting your analysis effectively, fostering better communication and data exploration for users.

1. **Tidyverse**

Tidyverse is a collection of interrelated R packages designed specifically for data science workflows. It emphasizes a “tidy data” philosophy and offers a consistent set of tools for data manipulation, visualization, and modelling.

Core tidyverse packages are:

* + ggplot2: A powerful grammar-based graphics approach to data wrangling, making tasks like filtering, summarizing, and transforming your data more intuitive and efficient.
  + tidyr: It provides tools for reshaping data into the “tidy” format preferred by other tidyverse packages, ensuring consistency and streamlined analysis.
  + readr: Offers functions for efficiently reading data from various file formats, making it easy to import your sales data from CSV, Excel, or other sources.
  + purrr: Provides functions for functional programming in R, useful for iterating over data structures and automating repetitive tasks within your sales data analysis.
  + tibble: A data frame implementation offering improved performance and readability compared to base R data frames, often used to store and manipulate your sales data.
  + stringr: A collection of functions for string manipulation, helpful for cleaning and preparing textual data related to your sales, such as product names or customer details.
  + forcats: Provides tools for working with categorical variables, including factors and ordered factors, which can be helpful for analyzing sales data categorized by product type, region, or other factors.

By leveraging tidyverse, we can effectively analyze and visualize your data, gaining valuable insights.

Official documentation: <https://www.tidyverse.org/>

1. **Reactable**

Reactable is a powerful R package that allows you to create interactive and visually appealing data tables, perfect for incorporating into dashboards built with R Markdown or Shiny applications. A close look at what reactable offers:

* + Rich interactivity: Reactable tables are not static displays of data.They provide users with the ability to:
    - Sort data by clicking on column headers, allowing for quick exploration of trends and patterns in your data.
    - Filter data based on specific criteria, enabling users to focus on subsets of information relevant to their interests.
    - Paginate through large datasets, ensuring smooth navigation even when dealing with extensive data.
  + Built-in Column Formatting: Reactable offers various options for formatting table columns, allowing you to present your sales data clearly and concisely. You can:
    - Apply custom number formatting to display sales figures in a user-friendly way.
    - Use color coding or other visual cues to highlight specific values or trends within your sales data.
  + Customizable Appearance: Reactable allows you to tailor the look and feel of your data tables to match your overall dashboard design. You can control aspects like font size, color scheme, and overall table styling, ensuring a visually cohesive presentation.
  + Expandable Rows and Nested Tables: Reactable offers advanced features like expandable rows and nested tables. This functionality can be useful for displaying hierarchical data structures within your sales analysis. For example, you could show overall sales figures for a product category and then allow users to expand a row to see detailed sales data for each individual product within that category.

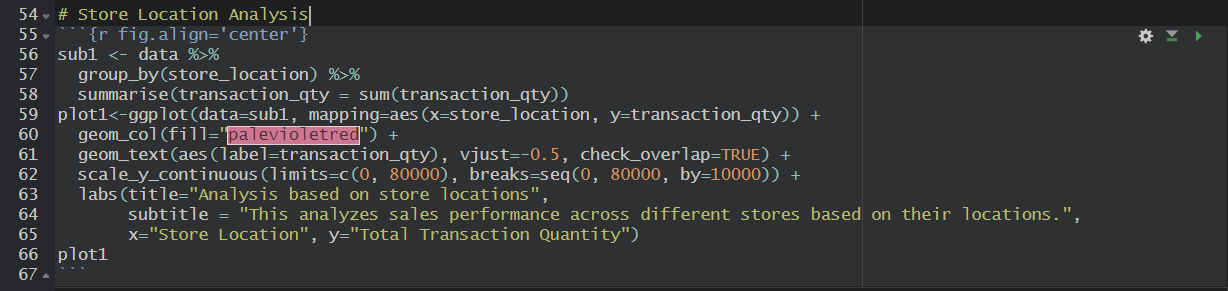
In summary, reactable empowers you to create informative and engaging data tables that enhance the user experience of the R Markdown documents or Shiny dashboards. By leveraging its interactive features and customization options, you can effectively communicate insights gleaned from your data analysis.

**Analysis on Dataset**

**Objective 1:** Store Sales Analysis

**Description:** This objective focuses on analyzing sales performance across different stores based on their locations.

**Specification:**

****

This code performs the following tasks:

1. Firstly, it creates a subset “sub1” by grouping the data by store\_location using dplyr package and then calculates the sum of transaction quantity based on each location using summarise() function.
2. Later, it creates a plot which showcases the pictorial view of sub1 using ggplot2 package. Store Location is represented in x-axis and Total Transaction Quantity in y-axis.
3. The labels of transaction quantity was represented using geom\_text() and customized the values of y-axis using scale\_y\_continuous().

Here’s the output:

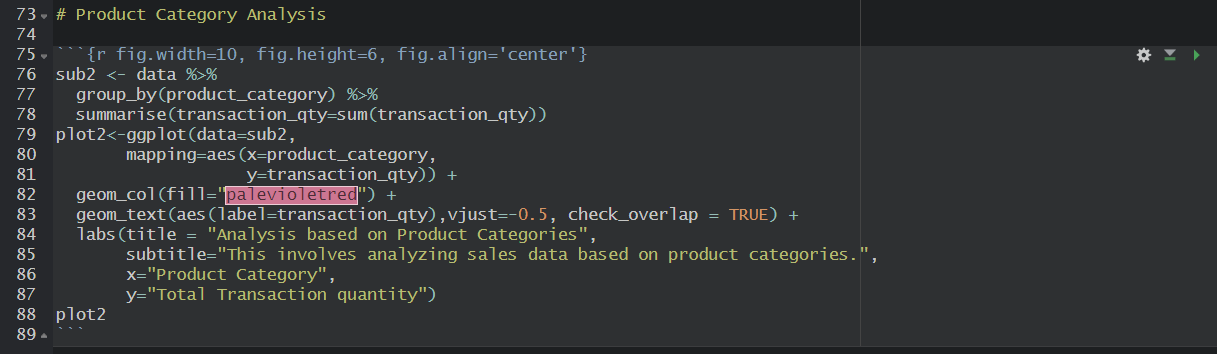
****

Overall, this represents the performance of sales based on store location in a column graph.

**Objective 2:** Product Category Sales Analysis

**Description:** This objective involves analyzing sales data based on product categories.

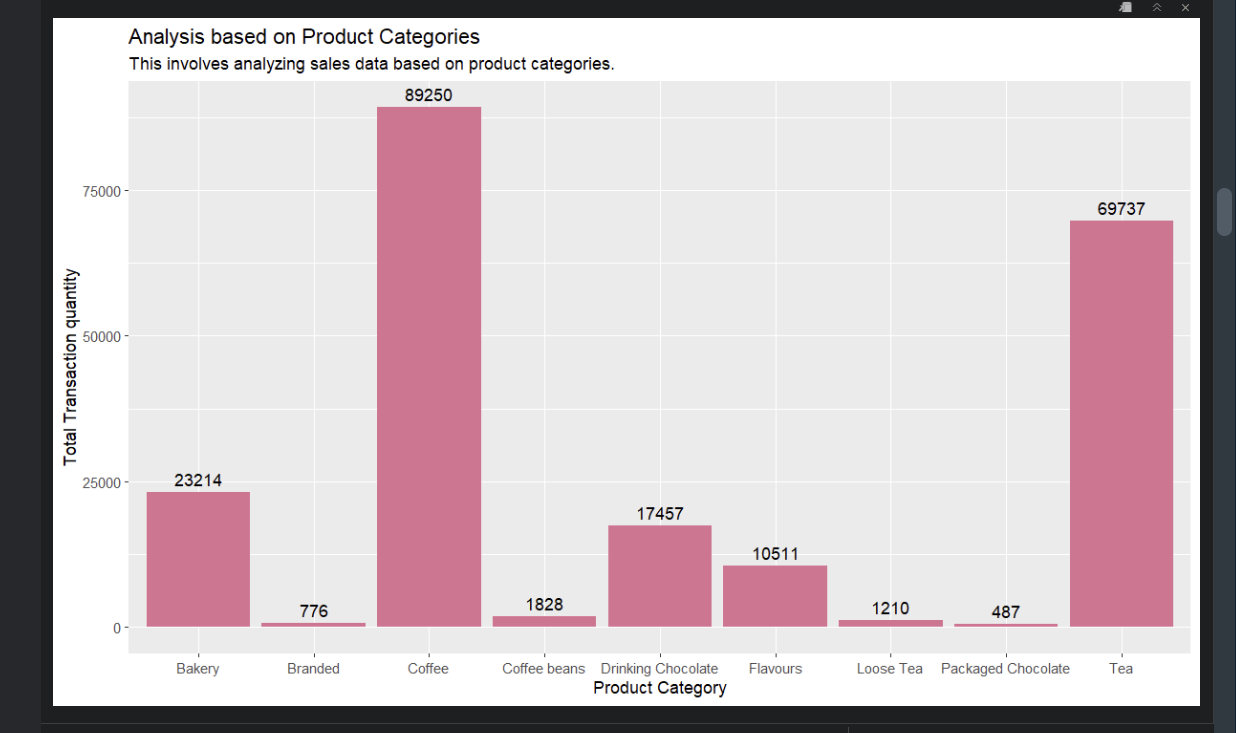
**Specification 1:**



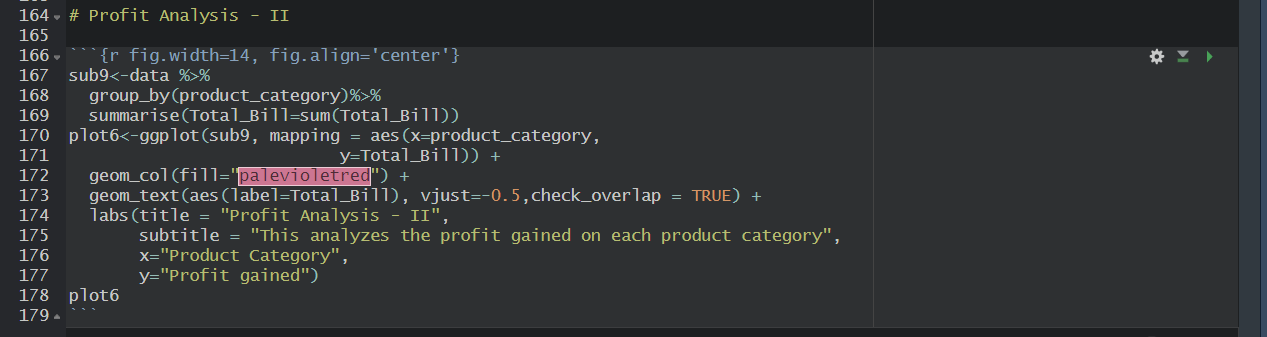
This code performs the following tasks:

1. Firstly, it creates a subset “sub2” by grouping the data by product\_category and then calculates the sum of transaction quantity based on product\_category using summarise() function from the dplyr package.
2. Later, it creates a plot of the sub2 using ggplot2 package where Product Category is represented on x-axis and Total Transaction Quantity is represented on y-axis.

Here’s the output:

Overall, this analysis tells us about the performance of sales based on product categories of all stores.

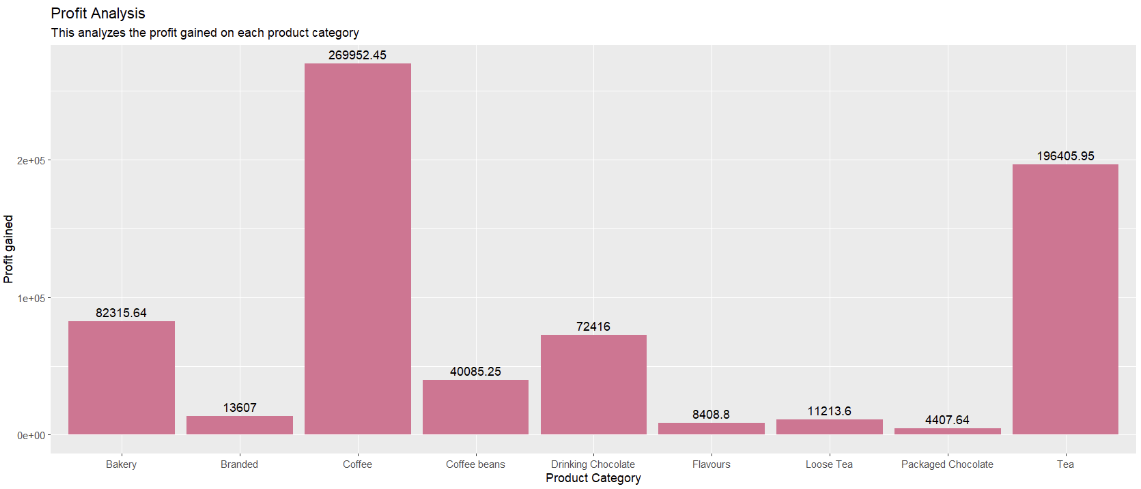
**Specification 2:**



This code performs the following tasks:

1. Firstly, this code creates a subset “sub9” which groups the data by product\_category and calculates sum of Total Bill using summarise() function in dplyr package.
2. Later, this creates a plot of sub9 using ggplot2 package where Product Category is represented on x-axis and Profit gained on y-axis.

Here’s the output:

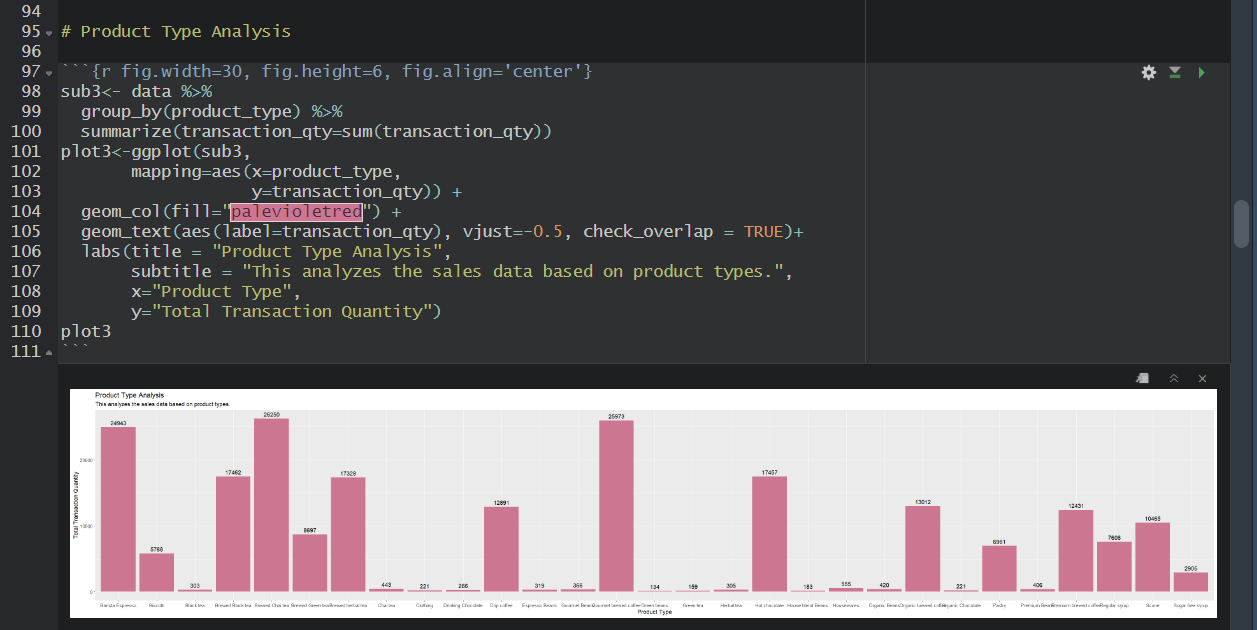


Overall, this analysis tells us about profit acquired on each product category from all store locations.

**Objective 3:** Product Type Sales Analysis

**Description:** This objective involves analyzing sales data based on product types.

**Specification 1:**



This code performs the following tasks:

1. Firstly, it creates a subset “sub3” by grouping the data by product\_type and then calculates the sum of transaction quantity using summarise() function in dplyr function.
2. Later, it creates a plot of subset “sub3” where Product type is represented on x-axis and Total Transaction Quantity is represented on y-axis using ggplot2 package.

Overall, this analysis gives the information about the performance of sales based on product type of all store locations.

**Specification 2:**



This code performs the following tasks:

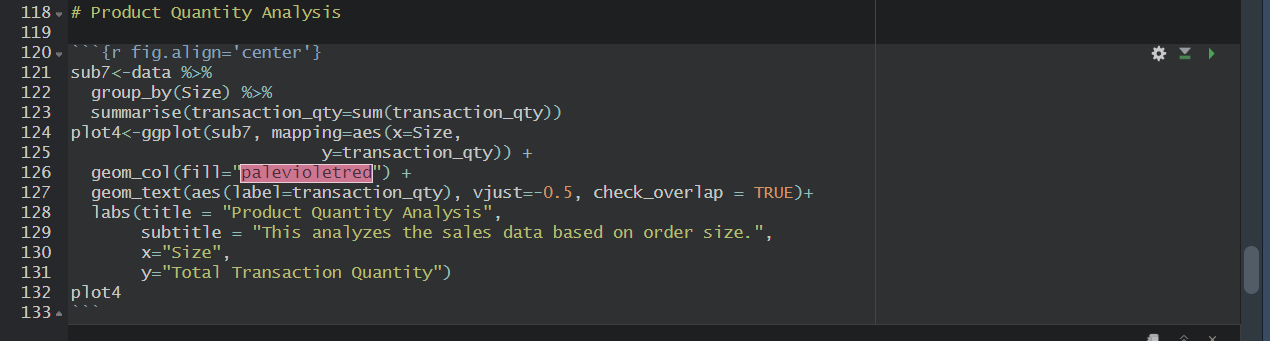
1. Firstly, this code creates a subset “sub8” by grouping the data by product\_type and calculates the Total\_Bill based on product\_type using summarise() from dplyr package.
2. Later, it creates a plot for subset “sub8” where Product Type is represented on x-axis and Profit gained on y-axis using ggplot2 package.

Overall, this analysis tells us about the profit gained on each product type from all store locations.

**Objective 4:** Quantity Analysis

**Description:** This objective focuses on analyzing sales data based on the quantity of products sold.

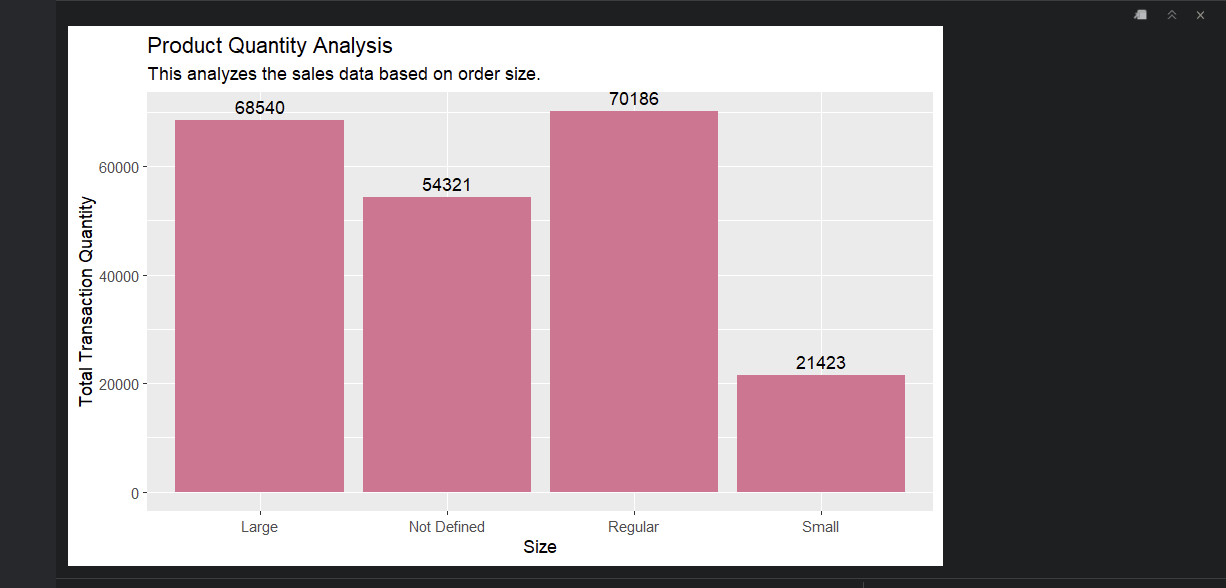
**Specification:**



This code performs the following tasks:

1. Firstly, this code creates a subset “sub7” by grouping the data by size and then calculates the quantity of products sold using summarise() from dplyr package.
2. Later, it creates a plot for subset “sub7” where Size of the product is represented on x-axis and Total Transaction quantity is represented on y-axis using ggplot2 package.

Here’s the output:

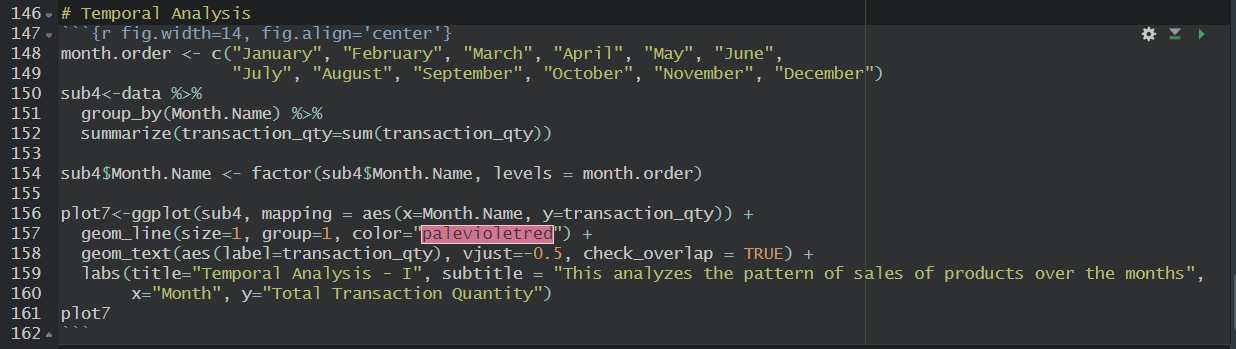


Overall, this analysis tells us about the performance of sales based on product quantity i.e. size of the product, from all store locations.

**Objective 5:** Temporal Analysis

**Description:** This objective involves analyzing sales data based on purchase date and time.

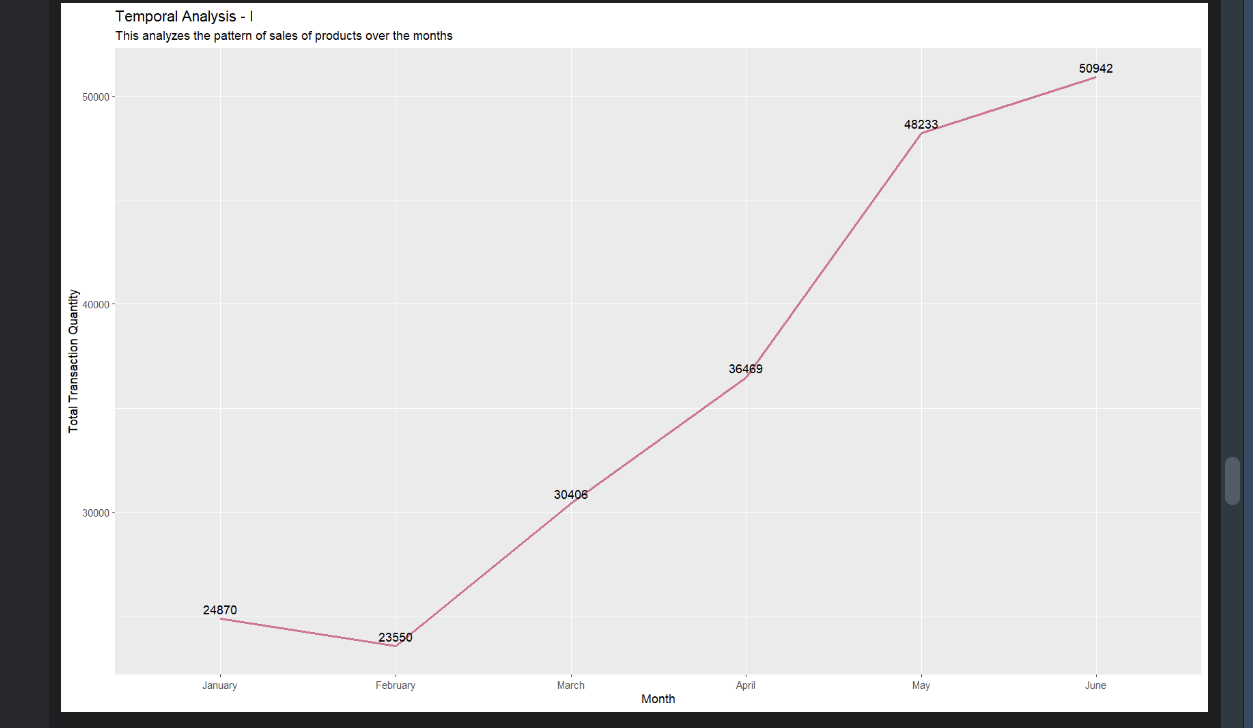
**Specification 1:**



This code performs the following tasks:

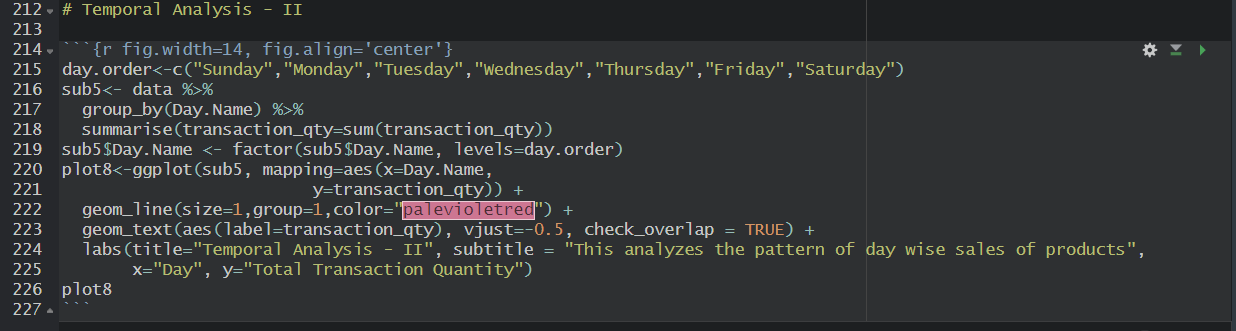
1. Firstly, it creates a subset “sub4” by grouping the data by Month.Name and then calculates the sum of transaction\_qty using summarise() in dplyr package.
2. Later, this creates a plot where the visualization is represented in trendline using geom\_line() from ggplot2 package. Month Name is represented on x-axis and Total Transaction Quantity is represented on y-axis.
3. The data contains the order of months in random order. So, to do analysis using clean data, a dataframe is created which has the correct order of month names and replaces the order of months of dataset with the dataframe created.

Here’s the output:



Overall, this analysis tells us about the trend of sales of products over months.

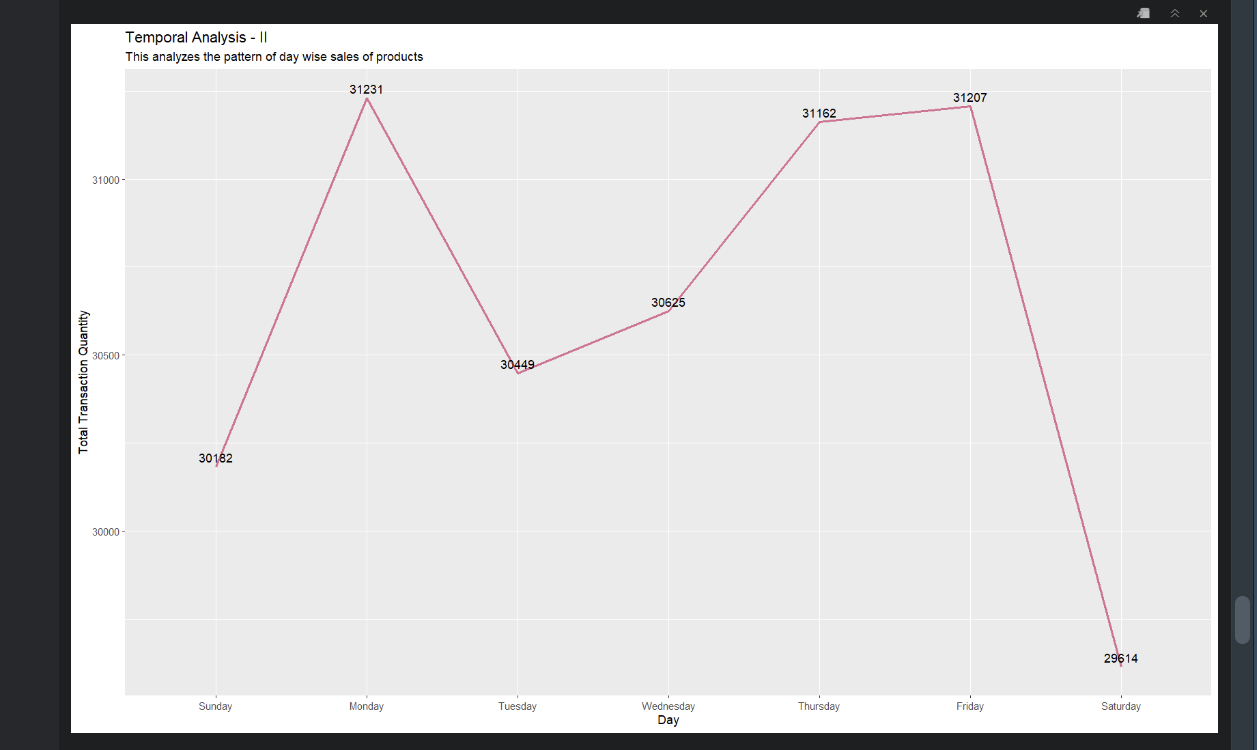
**Specification 2:**



This code performs the following tasks:

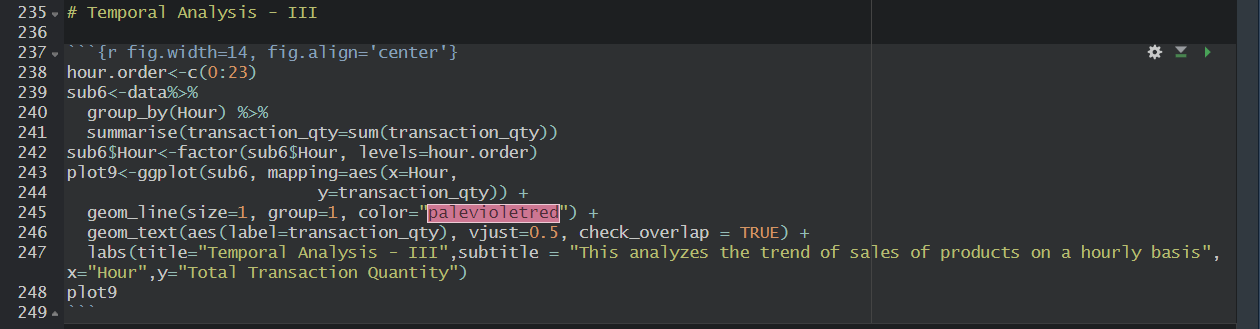
1. Firstly, it creates a subset “sub5” by grouping the data by Day.Name and then calculates the sum of transaction\_qty using summarise() in dplyr package.
2. Later, this creates a plot where the visualization is represented in trendline using geom\_line() from ggplot2 package. Day Name is represented on x-axis and Total Transaction Quantity is represented on y-axis.
3. The data contains the order of days in random order. So, to do analysis using clean data, a dataframe is created which has the correct order of day names and replaces the order of days of dataset with the dataframe created.

Here’s the output:



Overall, this analysis tells us about the trend of sales of products over days of a week.

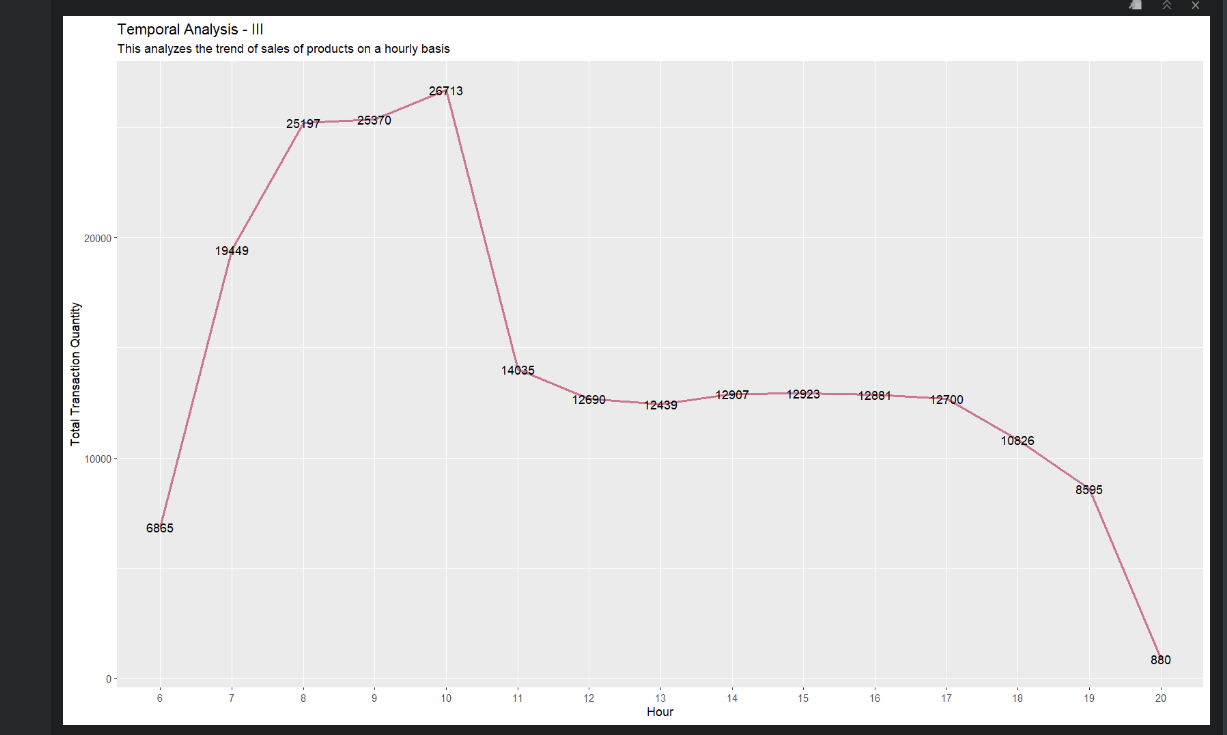
**Specification 3:**



This code performs the following tasks:

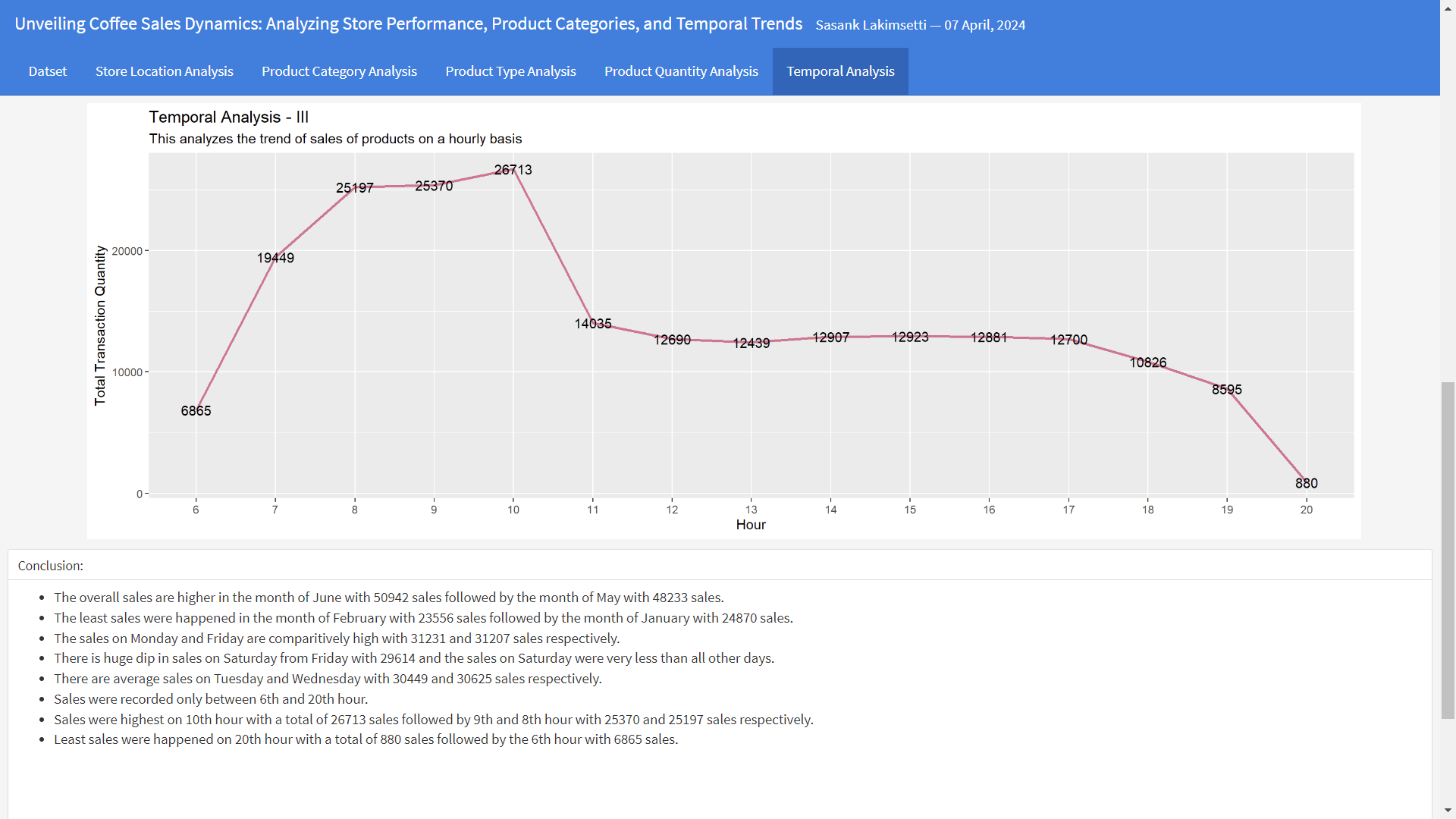
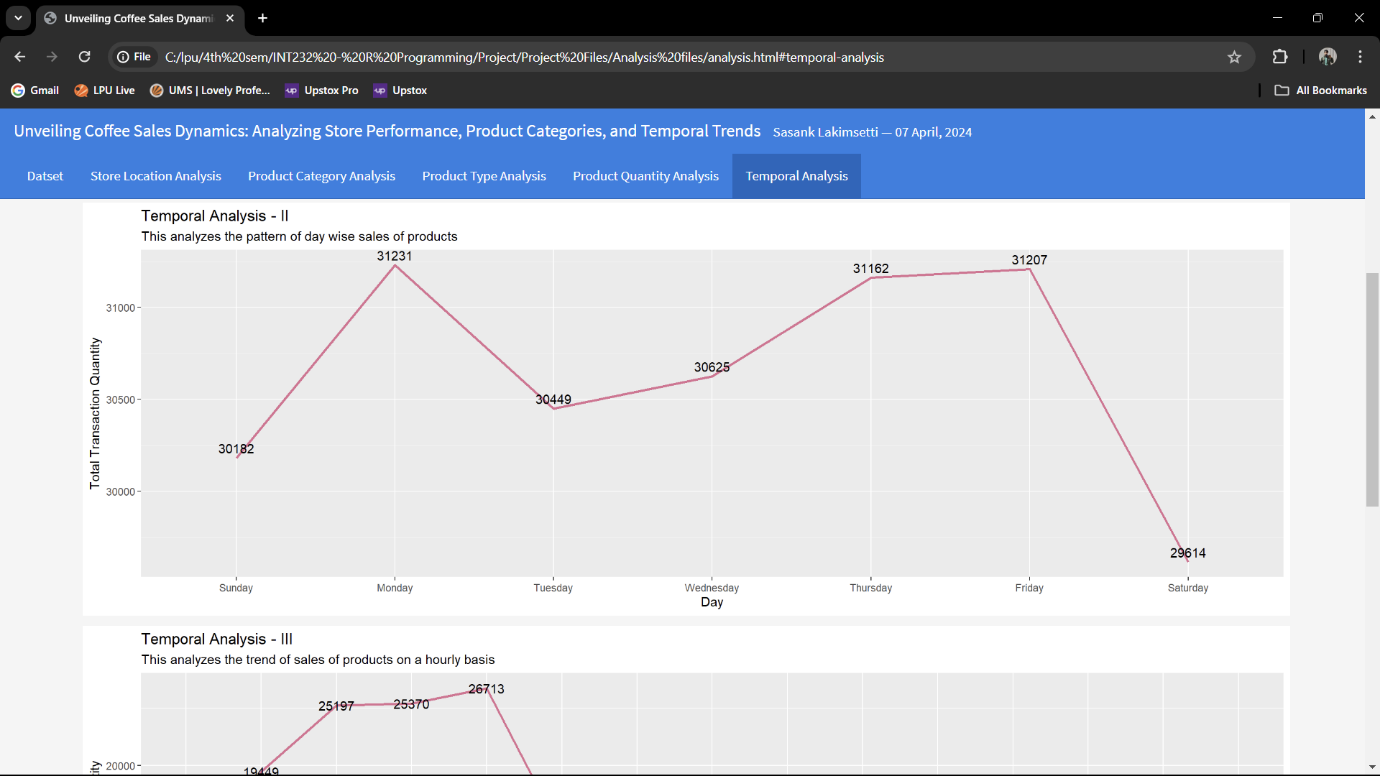
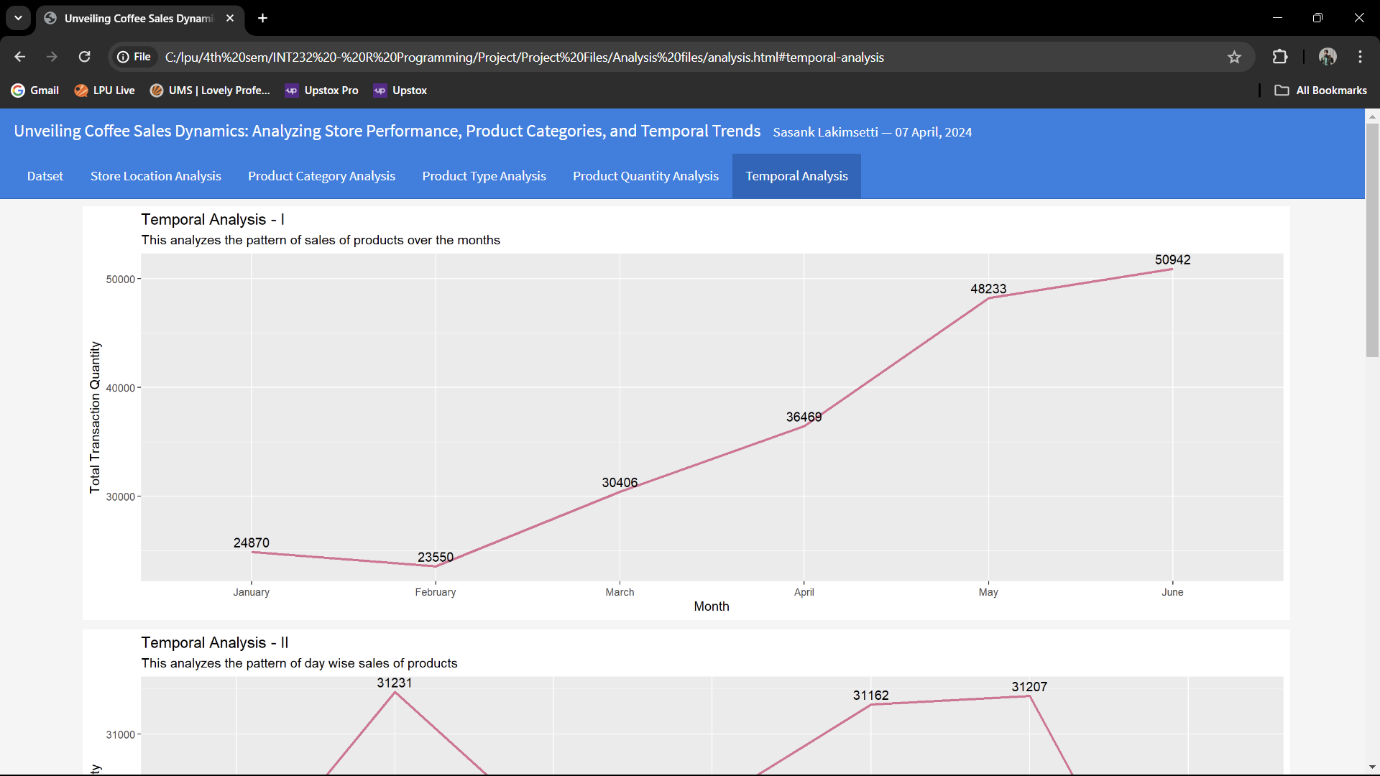
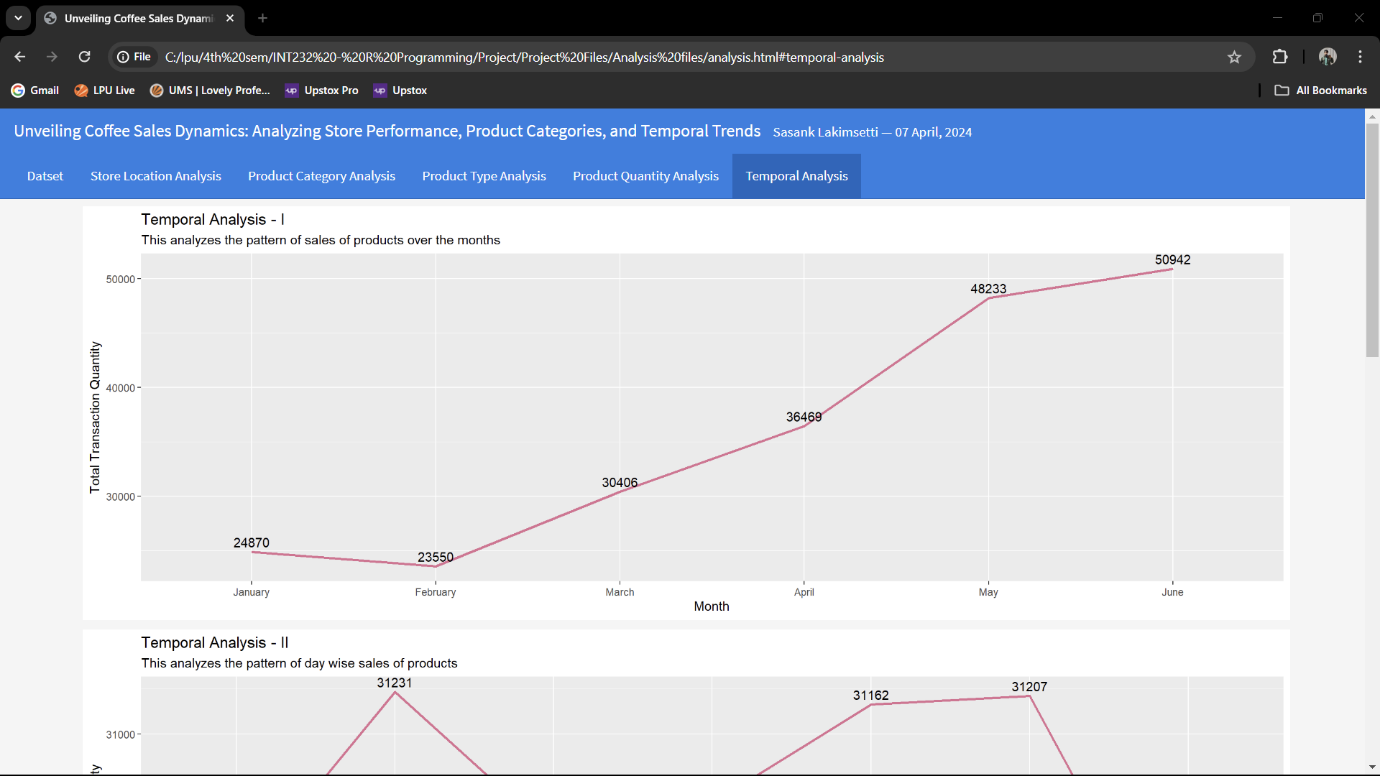
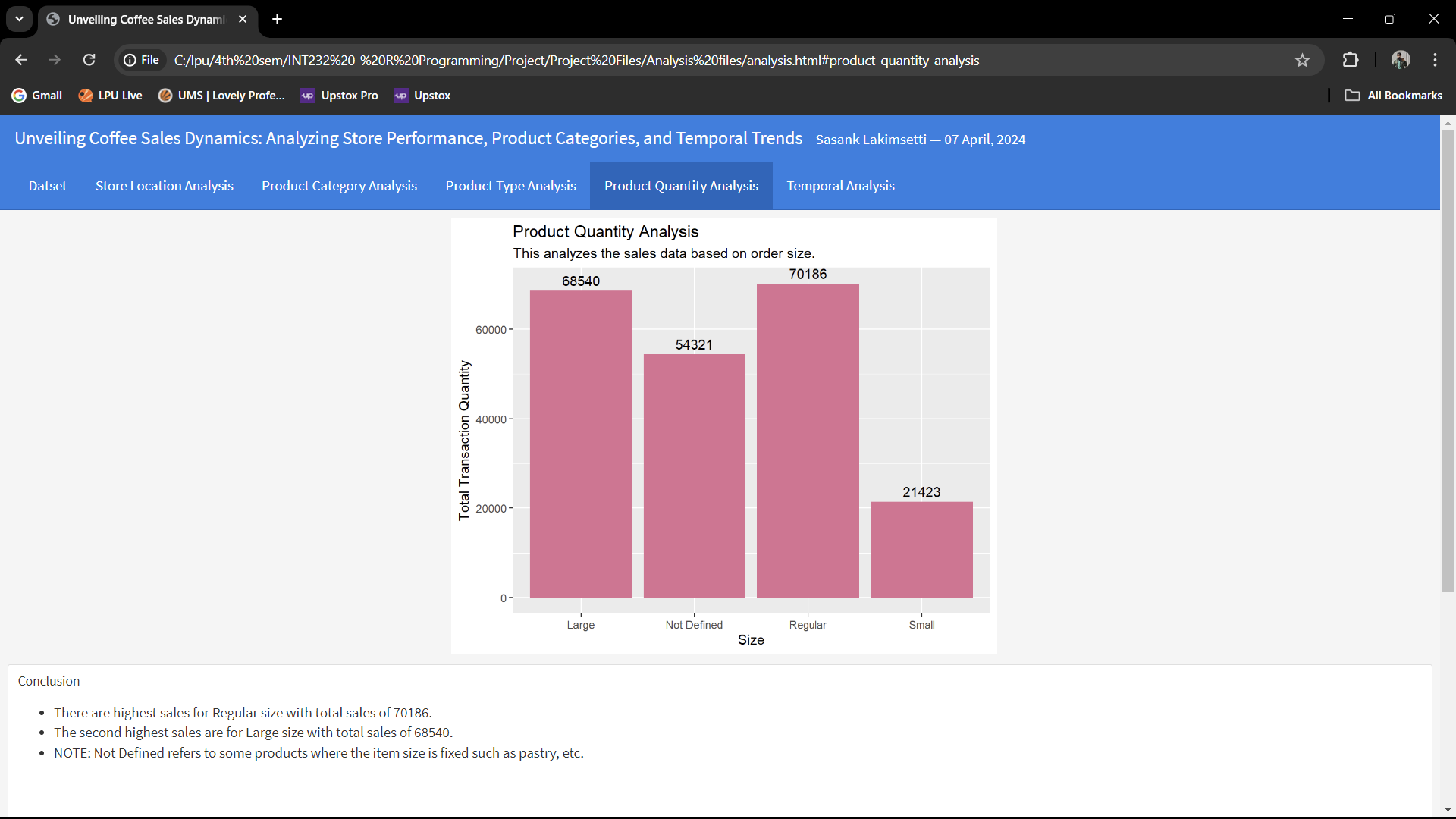
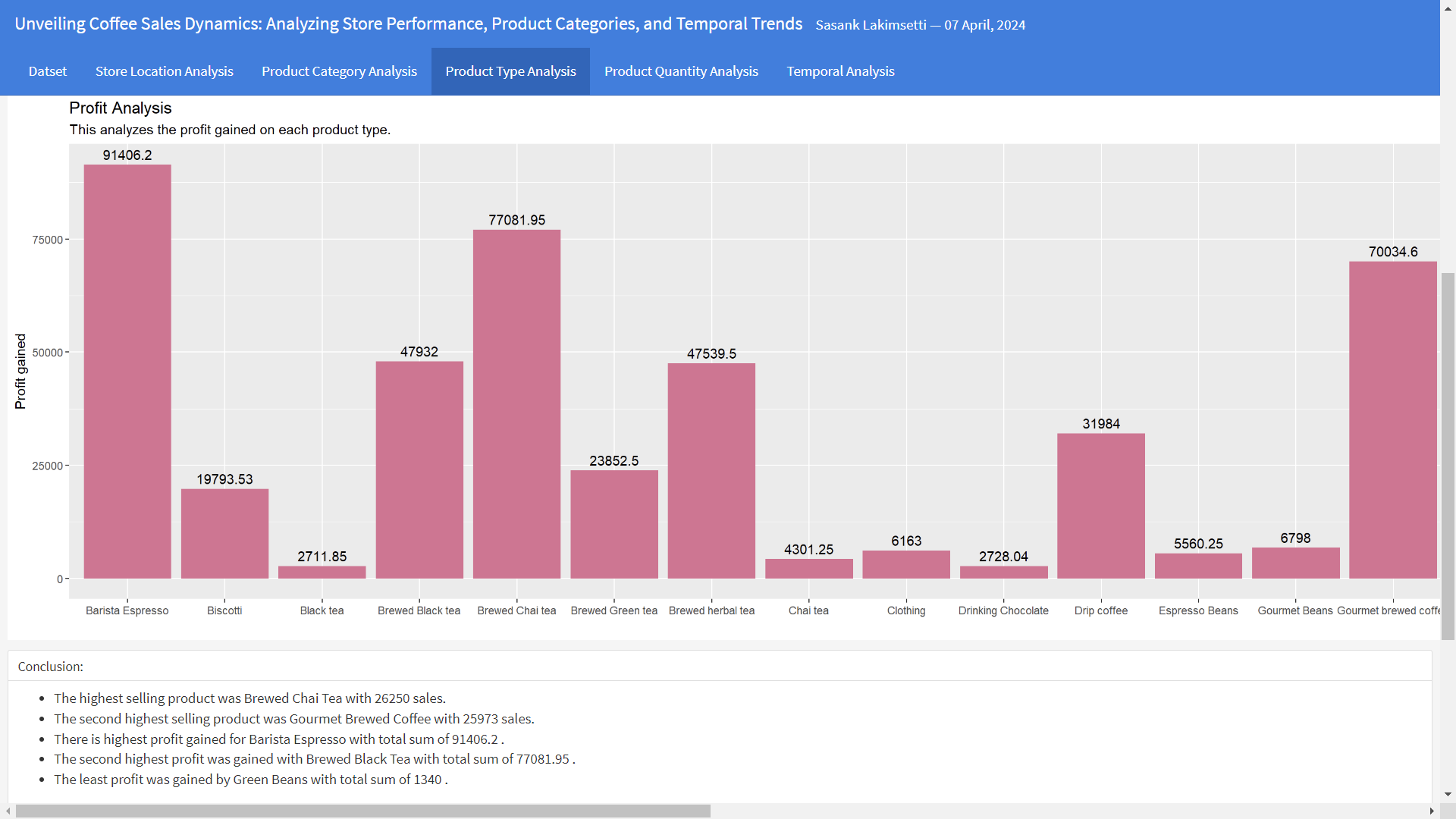
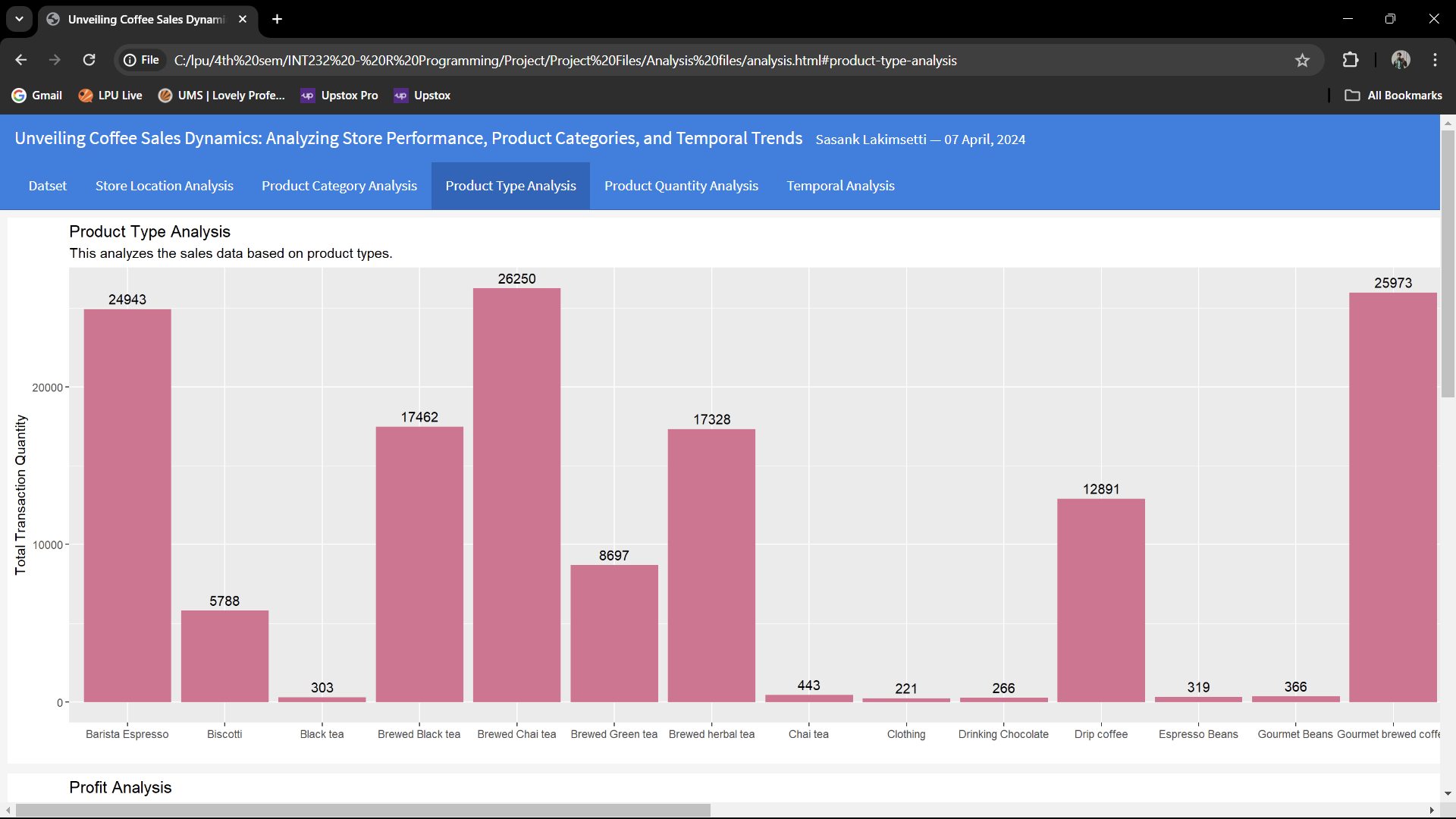
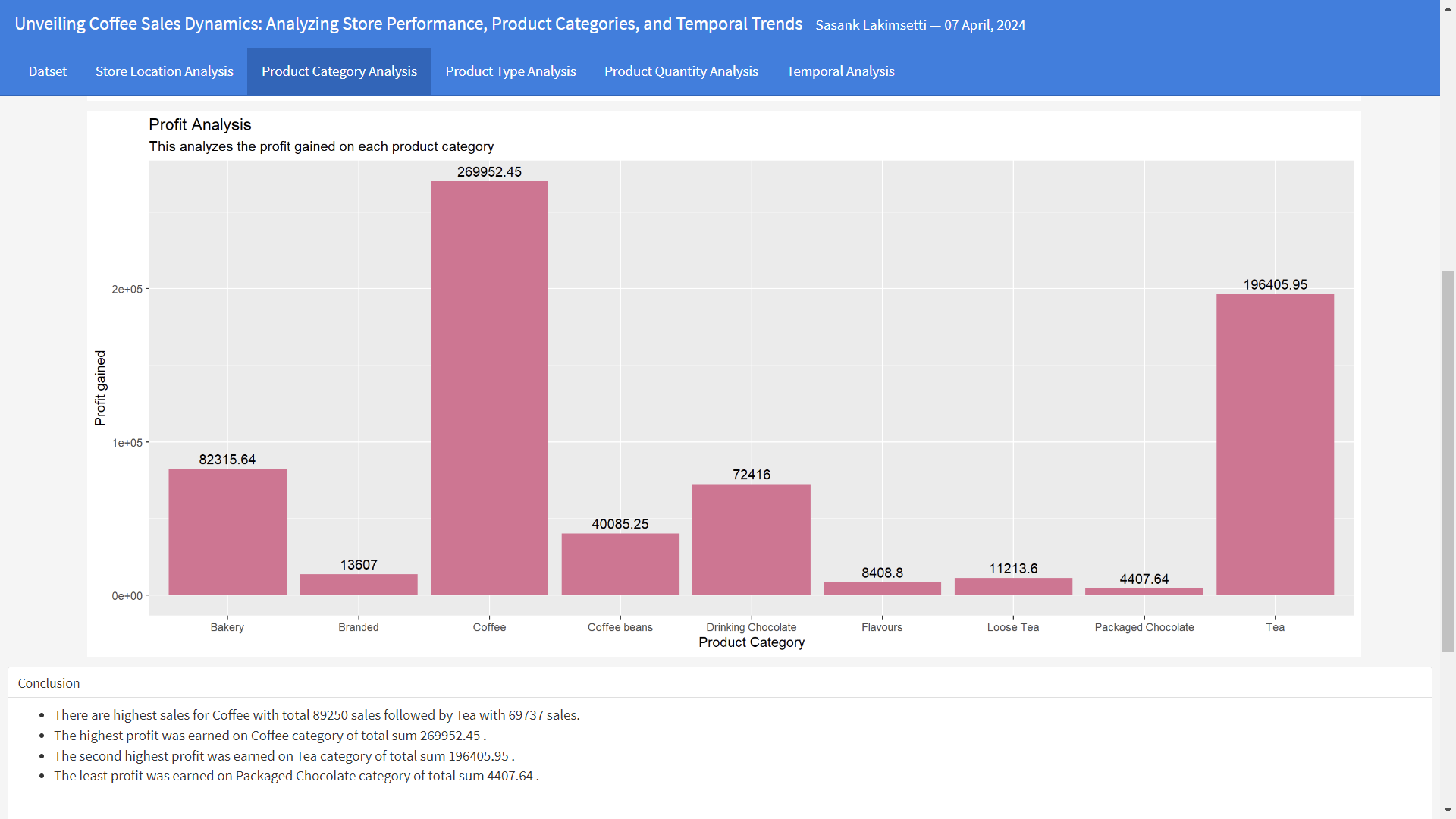
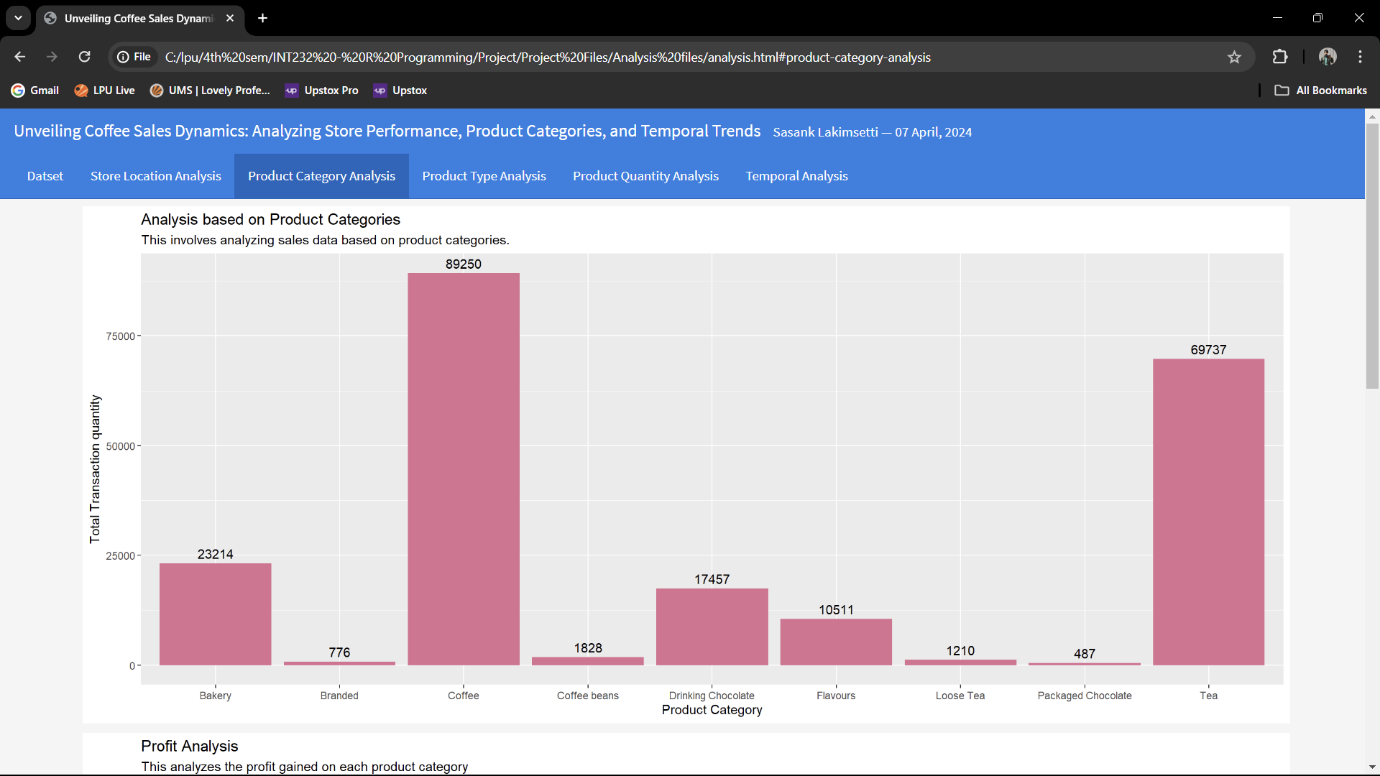
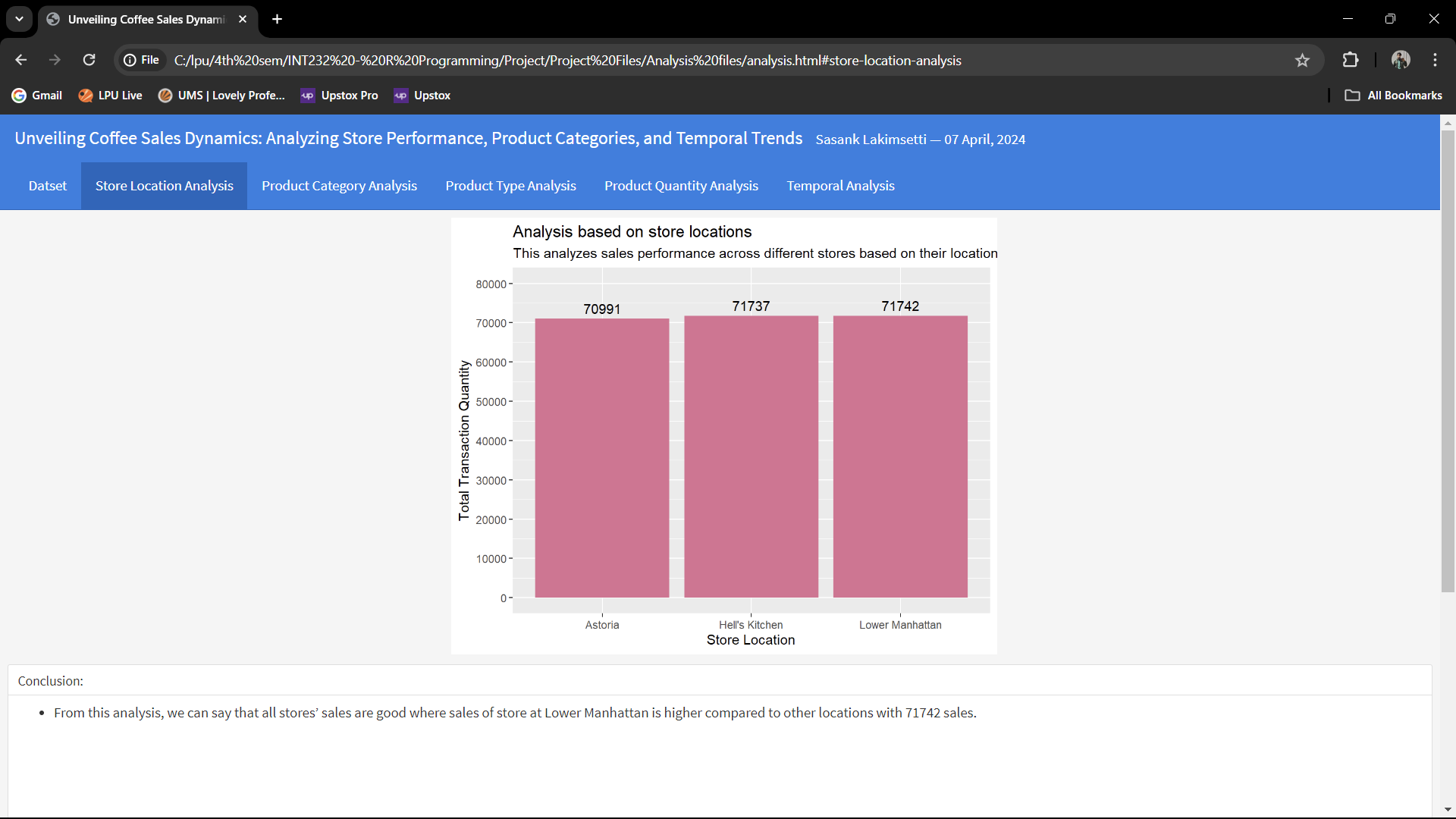
1. Firstly, it creates a subset “sub6” by grouping the data by Hour and then calculates the sum of transaction\_qty using summarise() in dplyr package.
2. Later, this creates a plot where the visualization is represented in trendline using geom\_line() from ggplot2 package. Hour is represented on x-axis and Total Transaction Quantity is represented on y-axis.
3. The data contains the order of hour in random order. So, to do analysis using clean data, a dataframe is created which has the correct order of hour values from 0 to 23 and replaces the order of hour values of dataset with the dataframe created.
4. Actual data consists of hour values from 6 to 20.

Here’s the output:



Overall, this analysis tells us about performance of sales on hourly-basis of all store locations.

**Final Dashboard**



The dashboard has been made using “flexdashboard” package which finally creates a html document which contains the dashboard analysis.

**Future Scope and Summary**

This R Markdown report has effectively presented a comprehensive analysis of Coffee Store Chain's sales performance during the first half of 2023. It leveraged the strengths of R programming and R Markdown to create an interactive dashboard that visualizes key metrics like quantity sold, profit margins, and sales trends.

**Future Scope:**

**Expand the Analysis**: This report can be extended to encompass a broader timeframe, allowing for comparisons across years or seasons. Additionally, incorporating external data sources like weather patterns or competitor activity could reveal further insights into sales trends.

**Advanced Data Analysis**: Techniques like machine learning could be employed to forecast future sales or identify potential customer segments. This would provide valuable information for optimizing marketing strategies and inventory management.

**Deployment and Sharing**: Consider deploying the dashboard as a Shiny application, making it easily accessible to stakeholders through a web interface. This would allow for real-time updates and broader data exploration by relevant decision-makers.

**Summary**:

By utilizing R programming and R Markdown, this report successfully delivered an interactive dashboard that empowers stakeholders with data-driven insights into Coffee Store Chain's sales performance. The key takeaways include:

* + Understanding the most popular products and potential areas for inventory optimization.
  + Identifying high-profit margins and opportunities to maximize revenue.
  + Recognizing sales trends over time and using them to inform marketing strategies.

Moving forward, the future scope outlined above presents exciting possibilities for further enriching the analysis and its practical applications. This data-driven approach can be instrumental in driving informed decision-making and propelling Coffee Store Chain towards continued success.

**References**

[1] <https://www.kaggle.com/datasets/divu2001/coffee-shop-sales-analysis>

[2] <https://rmarkdown.rstudio.com/lesson-12.html>

[3] <https://youtu.be/vWGGlMHkhyY?si=0---a9AUTum-PWwR>

[4] <https://youtu.be/bEhjiM8D7T0?si=0TmGvcaZJqueUjPp>

[5] <https://youtu.be/oB40BjRSu9A?si=mxE3ldpN2q05nhrq>

[6] <https://ggplot2.tidyverse.org/>

Project link : <https://github.com/sasanklakimsetti/Data-Analysis-2>